# HMM: The Learning Problem. Part II: Maximum Likelihood and the EM Algorithm Foundations

Sorin Istrail

Department of Computer Science
Brown University, Providence
sorin@cs.brown.edu

April 2, 2020

# Outline

# The Principle of Maximum- Likelihood

- The general prinicple of Maximum-Likelihood
- Suppose that we have $c$ data sets $\mathcal{D}_1 ... \mathcal{D}_c$ with the sample $\mathcal{D}_j$ haveing been drawn independently according to the probability distribution $p(x \mid w_j)$

- We say that such sample are i.i.d.-idependent and identically distributed random variables
- we assume that $p(x \mid w_j)$ has a **known parameter form**, and therefore determined uniquely by the value of its paramenter vector $\theta_j$
- For example, we might have $p(x \mid w_j) = N(\mu_j, \sigma_j)$ where $\theta_j$ is the vector of all components of $\mu_j, \sigma_j$.

# The Problem we want to solve

- **Notation**
- To show the dependence of of $p(x \mid w_j)$ on $\theta_j$ explicitly, we write $p(x \mid \theta_j)$
- **The Problem we want to solve**
- Use the information provided by the training samples to obtain good estimates for the unknown parameter vectors $\theta_1, ..., \theta_c$
- To simplify, assume that $\mathcal{D}_i$ give no information about $\theta_j, j \neq i$. Parameters are different classes are functionally different. And so we now have $c$ problems of the same form. So we will work with a generic one such data set $\mathcal{D}$.
- We use a set $\mathcal{D}$ of training samples drawn independently from the probability distribution $p(x \mid \theta)$ to estimate the unknown parameters vector $\theta$.

# The Maximum Likelihood Estimate

- Suppose $\mathcal{D}$ contains $n$ samples $x_1, \ldots, x_n$. Because the samples were drawn independently we have
- $$p(\mathcal{D} \mid \theta) = \prod_{k=1}^{n} p(x_k \mid \theta)$$
- $p(\mathcal{D} \mid \theta)$ viewed as a function of $\theta$ is the likelihood of $\theta$ with respect to $\mathcal{D}$
- The maximum-likelihood estimate of $\theta$ is, by definition, the value $\hat{\theta}$ that maximizes $p(\mathcal{D} \mid \theta)$
- Intuitively, this estimate corresponds to the value of $\theta$ that in some sense best agrees with or supports the actually observed training sample.

# Log-Likelihood maximization

- For analytical reasons, it is easy to work with the logarithm of the likelihood than with the likelihood itself, so we use the log-likelihood objective function
- Because the logarithm is monotonically increasing, the $\hat{\theta}$ that maximizes the log-likelihood also maximizes the likelihood
- If $p(\mathcal{D} \mid \theta)$ is a differentiable function of $\theta$, $\hat{\theta}$ can be found by standard differntial calculus methods

- If $\theta = (\theta_1, ..., \theta_r)^T$, let $\nabla_\theta$ be the **gradient operator**

$$\nabla_\theta = (\frac{\partial}{\partial \theta_1}, ..., \frac{\partial}{\partial \theta_r})^T$$

- Define $L(\theta)$ as the **log-likelihood function**

$$L(\theta) = \ln p(\mathcal{D} \mid \theta)$$

  and

- $$\hat{\theta} = \arg\max L(\theta)$$

- as the argument that Maximizes the log-likelihood; the dependence on $\mathcal{D}$ is implicit.

- We have by the independence condition

$$L(\theta) = \sum_{k=1}^{n} \ln p(x_k \mid \theta)$$

and

- $$\nabla_\theta L = \sum_{k=1}^{n} \nabla_\theta \ln p(x_k \mid \theta)$$

- This the necessary conditions for the maximum-likelihood estimate for $\theta$ can be obtained from the set of $r$ equations

$$\nabla_\theta L = 0$$

# The Expectation-Maximization (EM) Algorithm

- We extend now our application of maximum likelihood to permit **learning of parameters** governing a distributionfrom training points, some of which have **mising data** features.
- If there is no missing data, we can use maximum likelihood, i.e., find $\hat{\theta}$ that maximizes the log-likelihood $L(\theta)$.

- The basic idea of the EM algorithm is to iteratively estimate the likelihood given the data that is present.
- Consider a full sample $\mathcal{D} = \{x_1, ..., x_n\}$ of points taken from a single distribution. Suppose that some features are missing: so we can define for each sample point $x_k = \{x_{k_g}, x_{k_b}\}$
- i.e., contianing **“good”** features and the missing data as **“bad”** features.

- Let us separate the features in two classes $\mathcal{D}_g$ and $\mathcal{D}_b$, where $\mathcal{D} = \mathcal{D}_g \cup \mathcal{D}_b$
- Next we define the **Baum function**

$$\mathcal{Q}(\theta; \theta^i) = \mathcal{E}_{\mathcal{D}_b}(\ln p(\mathcal{D}_g, \mathcal{D}_b; \theta) \mid \mathcal{D}_g; \theta^i)$$

- known as the **Central Equation**
- where $\mathcal{Q}$ is a function of $\theta$ with the $\theta^i$ assumed fixed, and
- $\mathcal{E}_{\mathcal{D}_b}$ is the expectation operator computing the expected value marginalized over the missing features assuming $\theta^i$ are the "true" parameters describing the full distribution

- The **best intuition** behind the Central Equation in the EM algorithm is as follows:
- The parameter vector $\theta^i$ is the current best estimate for the full distribution
- $\theta$ is a candidate vector for an improved estimate

- Given such a candidate $\theta$, the right-had side of the central equation calculates the likelihood of the data including the unknown features $\mathcal{D}_b$ marginalized with respect to the current best distribution which is described by $\theta^i$

- Different such candidates will lead to different such likelihoods

- Our algorithm will select the best such candidates $\theta$ and call it $\theta^{i+1}$, the one corresponding to the greatest value of $\mathcal{Q}(\theta; \theta^i)$

```
Expectation-Maximization  (EM) Algorithm
BEGIN    Initiatlize theta powerto 0, epsilon, i=0
------
       DO i=i+1

          E step: Compute Q(theta; theta topower i)

          M step: theta topower {i+1} = arg max
                                    Q(theta, theta topower i)

       UNTIL Q(theta topower {i+1}; theta topower i) -
              Q((theta powerto i; theta topower {i-1}) <= e

       RETURN theta-hat = theta topower {i+1}
END
----
```

- The EM algorithm is most useful when the optimization of the $\mathcal{Q}$ function is simpler than the likelihood $L$.
- Most importantly, the algorithm guarantees that the log-likelihood of the good data (with the bad data marginalized) will increase monotonically.
- This is not the same as finding the particular values of the bad data that givess the maximum-likelihood of the full, complete data.